L3: Entry 4 of 4 File: USPT Dec 28, 1999

DOCUMENT-IDENTIFIER: US 6009392 A
TITLE: Training speech recognition by matching audio segment frequency of occurrence with frequency of words and letter combinations in a corpus

Abstract Text (1):
A method is provided which trains acoustic models in an automatic speech recognizer ("ASR") without explicitly matching decoded scripts with correct scripts from which acoustic training data is generated. In the method, audio data is input and segmented to produce audio segments. The audio segments are clustered into groups of clustered audio segments such that the clustered audio segments in each of the groups have similar characteristics. Also, the groups respectively form audio similarity classes. Then, audio segment probability distributions for the clustered audio segments in the audio similarity classes are calculated, and audio segment frequencies for the clustered audio segments are determined based on the audio segment probability distributions. The audio segment frequencies are matched to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches, and a textual corpus of words is formed based on the frequency matches. Then, acoustic models of the automatic speech recognizer are trained based on the textual corpus. In addition, the method may receive and cluster video or biometric data, and match such data to the audio data to more accurately cluster the audio segments into the groups of audio segments. Also, an apparatus for performing the method is provided.

Brief Summary Text (9):
In order to overcome the above problems, a method for training an automatic speech recognizer is provided. The method comprises the steps of: (a) inputting audio data; (b) segmenting the audio data to produce audio segments of the audio data; (c) clustering the audio segments into groups of clustered audio segments, wherein the clustered audio segments in each of the groups have similar characteristics and wherein the groups respectively form audio similarity classes; (d) calculating audio segment probability distributions for the clustered audio segments in the audio similarity classes; (e) determining audio segment frequencies for the clustered audio segments in the audio similarity classes based on the audio segment probability distributions; (f) matching the audio segment frequencies to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches; (g) forming a textual corpus of words based on the frequency matches; and (h) training acoustic models of the automatic speech recognizer based on the textual corpus.

Brief Summary Text (10):
Also, an apparatus for training an automatic speech recognizer is provided. The apparatus comprises: a receiver which inputs audio data; a segmenting device which is coupled to the receiver and which segments the audio data to produce audio segments of the audio data; an audio clustering device which is coupled to the segmenting device and which clusters the audio segments into groups of clustered audio segments, wherein the clustered audio segments in each of the groups have similar characteristics and wherein the groups respectively form audio similarity classes; a probability calculating device which is coupled to the audio clustering

device and which calculates audio segment probability distributions for the clustered audio segments in the audio similarity classes; a frequency determining device which is coupled to the probability calculating device and which determines audio segment frequencies for the clustered audio segments in the audio similarity classes based on the audio segment probability distributions; a frequency comparator which is coupled to the frequency determining device and which matches the audio segment frequencies to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches; a textual corpus generator which is coupled to the frequency comparator and which generates a textual corpus of words based on the frequency matches; and an acoustic model trainer which is coupled to the textual corpus generator and which trains acoustic models of the automatic speech recognizer based on the textual corpus.

Drawing Description Text (12):
FIG. 6 illustrates a flow chart for calculating probabilities of n-gram statistics.

Detailed Description Text (9):
After the strings of audio similarity classes are stored in the corpus, the probabilities of the segments and subsegments contained in the similarity classes are calculated, and the probabilities of the n-gram statistics (n=1, 2, 3, . . .) for the segments and subsegments contained in the similarity classes are calculated (step 108). An n-gram statistic relates to how many times a particular similarity class or a group of similarity classes is stored in the corpus of similarity classes. For example, a 1-gram statistic (or a 1-gram count) N.sub.C1 for the similarity class C1 is calculated by counting the number of times that the similarity class C1 is stored in the corpus. Also, 1-gram statistics N.sub.C2, N.sub.C3, N.sub.C4, etc. can be respectively calculated for each of the other similarity classes C2, C3, C4, etc. In the example described above, the corpus contains the similarity classes C1, C2, C3, C2, and C4. Therefore, the 1-gram statistics N.sub.C1, N.sub.C2, N.sub.C3, and N.sub.C4 respectively equal 1, 2, 1, and 1. The probability P.sub.Cx of a 1-gram statistic N.sub.Cx is calculated as a ratio of the 1-gram statistic N.sub.Cx to the total number of similarity classes N.sub.TOT contained in the corpus. In other words, P.sub.Cx =N.sub.Cx /N.sub.TOT. In the above example, the probability P.sub.C1 of the 1-gram statistic N.sub.C1 equals (N.sub.C1 /N.sub.TOT)=(1/5)=0.2. The probabilities P.sub.C2, P.sub.C3, and P.sub.C4 of the 1-gram statistics N.sub.C2, N.sub.C3, and N.sub.C4 respectively equal 0.4, 0.2, and 0.2.

Detailed Description Text (10):
A 2-gram statistic is similar to a 1-gram statistics except that it determines the number of times that a pair of similarity classes (e.g. C1 and C2) is contained in a corpus. Also, the probability of a 2-gram statistic is calculated in a manner which is similar to the manner in which the probability of a 1-gram statistic is calculated. An example of calculating the n-gram statistics and the probabilities of the n-gram statistics is described in more detail below in conjunction with FIG. 6.

Detailed Description Text (11):
After the probabilities are calculated in step 108, the frequencies of the segments and the n-gram segment statistics are matched with known frequencies and n-gram statistics for letters, combinations of letters, and words (step 109). The known frequencies may be obtained from large textual corpuses by using methods which are similar to those described in Lalit R. Bahl, et al., A Maximum Likelihood Approach to Continuous Speech Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, No. 2 (March 1983). Such reference is incorporated herein by reference.

Detailed Description Text (12):
To illustrate an example of how the known frequencies and n-gram statistics are

obtained, assume that an ASR is used to interpret acoustic data from a broadcast news program. In such instance, a news vocabulary of known words may be created based on the words contained in printed news media (e.g. newspapers, magazines, the internet, etc.). Since the total number of words contained in the various printed news media (i.e. the various textual corpuses) is very large, the news vocabulary may be created by selecting a certain number (e.g. 65,000) of the most frequently used words in the textual corpus. Thus, even though the acoustic data from the broadcast news program does not identically correspond to the words contained in the news vocabulary, the distribution probabilities and the n-gram statistics of the spoken words in the news program and the words in the news vocabulary are roughly the same because they relate to the same subject matter.

Detailed Description Text (13):
Once the frequencies of the segments and statistics have been matched with known segments and statistics in step 109, the best match of segments (and/or subsegments) to words (and/or subwords) which provides the best match of n-gram segment statistics and n-gram statistics for letters, combinations of letters, and words is determined (step 110). An example of how the best match is determined is described below in conjunction with FIG. 4 which illustrates how segments are matched to words by matching the probability distributions of the segments and words. The word or words that constitute the best match are compared with the corpus of audio similarity classes formed in step 107, and the word or words which match the similarity classes are used to form a corpus of words (step 111). In other words, by finding the best match between the frequencies of the segments and the known frequencies S and between the n-gram statistics for the segments and the n-gram statistics for letters, combinations of letters, and words, a string of similarity classes is mapped into a string of words by mapping each similarity class into a corresponding word. Then, the string of words are stored in a corpus of words.

Detailed Description Text (22):
FIG. 2a illustrates more detailed procedures that may be used in various steps shown in FIG. 1. In FIG. 2a, step 201 corresponds to step 103, step 202 corresponds to step 104, steps 203-207 correspond to step 105, and step 208 corresponds to step 107. As shown in the figure, audio signals are sampled at a frequency rate and a fast Fourier transform operation is applied to the sampled signals to produce strings of frames of audio data (step 201). The strings of frames are fragmented into all possible segments having lengths that are based on the average lengths of words, subwords, and phonemes and based on the probability distributions of the words, subwords, and phonemes (step 202). The probability distribution of a word is computed as a ratio of the number of times that the word appears in a textual corpus to the total number of words in the corpus. Similarly, the probability distribution of a subword or phoneme can be computed as a ratio of the number of times the word or phoneme appears in a textual corpus to the total number of subwords or phonemes in the corpus. The fragmentation process in step 202 should preferably begin with several of the most frequent lengths of phonemes and words.

Detailed Description Text (28):
If the clustering has not been adequately refined, a stock of similarity classes of the matched segments is formed and stored (step 207), the procedure returns to step 202, and the stock of similarity classes is used further refine the fragmentation of the frames of acoustic data into segments. Also, the stock of similarity classes can be used to ref ine the manner in which the segments are matched with each other. For instance, smaller subdivisions of segments that belong to each similarity class can be compared with each other and matched among themselves within each set of similarity classes. Also, PCM data can be produced in step 201 for smaller intervals d for sampling the audio data, and refined acoustic strings fragmented in step 202 can be matched based on acoustic strings and clusters which have already been matched in a previous iteration of the routine. If the clustering has been adequately refined (step 206), all of the similarity classes obtained in

step 205 are collected and stored in a corpus (step 208). Furthermore, the similarity classes are stored in the corpus in an order that corresponds the order of their context in the acoustic data from which they were produced.

Detailed Description Text (29):
The ordered similarity classes form strings of clusters (e.g. strings of words in a corpus of words) and are aligned with the acoustic data with respect to time. Since a set of all of the different similarity classes is stored in the stock of similarity classes in step 207, a vocabulary of symbols (i.e. clusters) is obtained in step 207, and textual data for different levels is composed from these symbols in the corpus generated in step 208. Also, the symbols are marked with their respective lengths in order to preserve information reflecting their dominance and the hierarchy of the clusters (i.e. similarity classes). For example, the symbols may be marked (C1, C2, C3 . . .) for the level 1 lengths and may be marked (CC1, CC2, . . .) for level 2 lengths in the corpus of similarity classes formed in step 208.

Detailed Description Text (31):
A more detailed description of how the procedure in step 108 of FIG. 1 is performed will be described below. In step 108, the probabilities of the segments and subsegments stored in the corpus formed in step 208 are calculated, and the probabilities of the n-gram statistics for the segments and subsegments are calculated. The manner in which the probabilities of the n-gram statistics are generated is similar to the manner in which probabilities for n-gram statistics are generated from usual textual corpuses to train a language model. (A language model is synonymous with the n-gram statistics for words). A more detailed discussion of language models can be found in the reference Lalit R. Bahl, et al., A Maximum Likelihood Approach to Continuous Speech Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, No. 2 (March 1983) which was mentioned above. Such reference is incorporated herein by reference.

Detailed Description Text (32):
Before describing a specific embodiment of the procedure used in step 108, some elementary concepts will first be described. N-gram statistics contain counts for all n-tuples (n=1, 2, 3, . . .) of different symbols from the stock of similarity classes obtained in step 207. An "In-tuple" is a set of symbols which contains "n" symbols. For example, a 1-tuple may contain the symbol S1, a 2-tuple may contain the symbols S1 and S2, and a 3-tuple may contain the symbols S1, S2, and S3. For example, the sentence "A dog ate my homework" can be considered to be a set of symbols (i.e. words). The 1-tuples of words contained in the sentence are: {A}, {dog}, {ate}, {my}, and {homework}. Some of the 2-tuples of words contained in the sentence comprise: {A dog}, {dog ate}, {ate my}, and {my homework}. Some of the 3-tuples of words contained in the sentence include: {A dog ate}, {dog ate my}, and {ate my homework}. These statistics are collected separately for symbols corresponding to different levels, and the counts are used to estimate the probabilities that various strings contain certain symbols for the different levels.

Detailed Description Text (35):
FIG. 6 illustrates an example of a method in step 108 for calculating the n-gram statistics for the similarity classes of segments and subsegments and for calculating the probabilities of the n-gram statistics for the similarity classes. First, the similarity classes contained in the corpus formed in step 107 are matched with the similarity classes contained in the vocabulary formed in step 106 in order to identify the classes in the corpus which are contained in the vocabulary (step 600). Also, classes which are contained in the corpus that are not contained in the vocabulary are identified with an "unknown" class symbol C.sub.u. Then, a counter generates a 1-gram statistic (i.e. 1-gram count) N.sub.Cx for each class C.sub.x in the corpus and generates a 2-gram count N.sub.CxCy for each pair of classes C.sub.x and C.sub.y in the corpus. Such process is continued until the

n-gram counts N.sub.Cx . . . Cn for the classes C.sub.x to C.sub.n in the corpus have been generated (steps 601.1, 601.2, . . . , 601.n). Then, the 1-gram counts N.sub.Cx, 2-gram counts N.sub.CxCy, . . . , and n-gram counts C.sub.Cx . . . Cn are stored (steps 602.1, 602.2, . . . , and 602.n). The probability P(C.sub.x) of the 1-gram statistic N.sub.Cx for each class C.sub.x is computed by dividing the statistic N.sub.Cx by the total number of counts N.sub.TOT of all of the classes (step 603.1). The probabilities of the 2-gram statistics through the n-gram statistics are calculated in a similar manner (steps 603.2 to 603.n).

Detailed Description Text (36):
After the various probabilities in step 108 are calculated, the frequencies of the segments and n-gram segment statistics are matched with known frequencies and n-gram statistics in step 109. Then, in step 110, the best match of segments (and/or subsegments) to words (and/or subwords) that provides the best match of n-gram segment statistics and n-gram statistics for letters, combinations of letters, and words is determined. An example of how the best match is determined is described below in conjunction with FIG. 4.

Detailed Description Text (37):
First, various cluster probability distributions are stored (step 401), and various word probability distributions are stored (step 402). Then, a set of cluster probability distributions is matched with a set of word probability distributions using some distance metric (e.g. Kulback distance) (step 403). Pairs of cluster and word probability distributions that have small distances between each other are identified as matched pairs (step 404), and the matched pairs are used to build a multi-value map which is used to convert symbols in the similarity classes to words (step 405). This multi-value map may relate symbols to several words. For example, matching frequencies of symbols and words allows some symbols S={C1, C2, C3, . .} to be matched with some words T={W1, W2, W3, . . . } with some possible collisions. (A "collision" defines the situation in which several different similarity classes are mapped to the same word). This map is used to match further distributions in step 403 to produce a refined map for converting symbols of the similarity classes into words with a fewer number of collisions. For example, the matched symbols and words from the groups S and T can be used to match distributions of symbols and words that already correspond to the matched symbols and words from the groups S and T. This extended set of the pair of matched distributions allows more symbols to be matched with more words to further reduce the number of collisions. The above procedure is repeated until the number of collisions is reduced to a minimum and a one-to-one correspondence map of symbols into words is constructed (step 406). Then, the map is stored as an optimal correspondence map (step 407). Additional descriptions of various techniques for encoding symbols into words using known distributions can be found in Deavors, C. A. ans Kruh, L. "Machine Cryptography and Modern Cryptoanalysis", (Dedham, Mass.: Artech House 1985). Such reference is incorporated herein by reference.

Detailed Description Text (38):
An illustrative embodiment of an apparatus which uses the method shown in FIGS. 1 and 1A to train an ASR is shown in FIG. 5. As illustrated in FIG. 5, the apparatus includes an audio data recorder 501, a time stamper 502, a frame generator 503, a segmentation module 504, an audio clustering module 505, a first vocabulary memory 506, an audio similarity classes mapping module 507, a probability and n-gram statistics calculation module 508, first and second matching modules 509 and 510, a word corpus creation module 511, an ASR training module 512, a language model n-gram statistic module 518, and a second vocabulary memory 519.

Detailed Description Text (41):
The probability and n-gram statistics calculation module 508 inputs the corpus of audio similarity classes from the mapping module 507 and inputs the similarity classes from the memory 506. Then, the module 508 calculates the probabilities of the segments and subsegments contained in the similarity classes and calculates the

probabilities of the n-gram statistics for the segments and subsegments contained in the similarity classes.

Detailed Description Text (42):
The first matching module 509 inputs the frequencies of the segments and the n-gram segment statistics from the calculation module 508 and inputs known frequencies and n-gram statistics for letters, combinations of letters, and words from the n-gram statistic module 518. Then, the frequencies and n-gram statistics from the calculation module 508 are matched with the known frequencies and n-gram statistics from the module 518.

Detailed Description Text (43):
The second matching module 510 inputs the similarity classes from the first vocabulary memory 506, inputs word data from the second vocabulary memory 519, and inputs the results of the matching operation performed by the first matching module 509. Then, the matching module 510 determines the best match of segments (and/or subsegments) to words (and/or subwords) which provides the best match of n-gram segment statistics to n-gram statistics for letters, combinations of letters, and words.

CLAIMS:

1. A method for training an automatic speech recognizer, comprising the steps of:

(a) inputting audio data;

(b) segmenting said audio data to produce audio segments of said audio data;

(c) clustering said audio segments into groups of clustered audio segments, wherein said clustered audio segments in each of said groups have similar characteristics and wherein said groups respectively form audio similarity classes;

(d) calculating audio segment probability distributions for said clustered audio segments in said audio similarity classes;

(e) determining audio segment frequencies for said clustered audio segments in said audio similarity classes based on said audio segment probability distributions;

(f) matching said audio segment frequencies to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches;

(g) forming a textual corpus of words based on said frequency matches; and

(h) training acoustic models of said automatic speech recognizer based on said textual corpus.

9. The method as claimed in claim 8, wherein said step (d) further comprises the steps of:

(d3) determining n-gram statistics for each of said clustered audio segments in said audio segment corpus, wherein said n-gram statistics are based on a number of instances that a particular one of said clustered audio segments is contained in said audio segment corpus and are based on a number of instances that said particular one of said clustered audio segments is associated with at least another clustered audio segment in said audio segment corpus.

10. The method as claimed in claim 9, wherein said step (d) further comprises the step of:

(d4) calculating n-gram probability distributions of said n-gram statistics.

11. The method as claimed in claim 9, wherein said step (e) comprises the steps of:

(e1) determining n-gram segment statistic frequencies for said clustered audio segments based on said n-gram statistics for the clustered audio segments; and

(e2) determining said audio segment frequencies for said clustered audio segments based on said audio segment probability distributions.

12. The method as claimed in claim 11, wherein said step (f) comprises the steps of:

(f1) matching said audio segment frequencies to said known audio segment frequencies; and

(f2) matching said n-gram segment statistic frequencies to known n-gram statistic frequencies for said at least one of letters, combination of letters, and words.

13. The method as claimed in claim 12, wherein said step (f) further comprises the steps of:

(f3) determining a best n-gram match of said n-gram segment statistic frequencies to said known n-gram statistic frequencies; and

(f4) determining a best segment match of said audio segment frequencies to said known audio segment frequencies based on said best n-gram match, wherein said best n-gram match and said best segment match at least partially constitute said frequency matches.

14. The method as claimed in claim 13, wherein said step (g) comprises the steps of:

(g1) comparing said audio segments corresponding to said audio segment frequencies of said best segment match with said audio similarity classes formed in said step (c);

(g2) determining which of said audio segments constitute matching audio segments that match said audio similarity classes; and

(g3) forming said textual corpus by using said matching audio segments, wherein said textual corpus maps certain audio similarity classes to certain audio segments.

23. A method for training an automatic speech recognizer, comprising the steps of:

(a) storing audio data and time stamping said audio data when said audio data is stored;

(b) sampling said audio data to produce sampled audio data and converting said sampled audio data into a string of frames of audio data;

(c) segmenting said string of frames to produce audio segments;

(d) comparing said audio segments with each other to determine audio similarities among said audio segments and clustering said audio segments into groups of clustered audio segments based on said similarities, wherein said groups respectively form audio similarity classes;

(e) storing at least some of said audio similarity classes in a vocabulary of comprehensive audio similarity classes;

(f) forming an audio segment corpus of said clustered audio segments in said audio similarity classes, wherein said clustered audio segments are ordered in said audio segment corpus in accordance with an order in which said audio data is stored in said step (a);

(g) calculating audio segment probability distributions for said clustered audio segments in said audio segment corpus;

(h) determining audio segment frequencies for said clustered audio segments in said audio segment corpus based on said audio segment probability distributions;

(i) matching said audio segment frequencies to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches;

(j) forming a textual corpus of words based on said frequency matches; and

(k) training acoustic models of said automatic speech recognizer based on said textual corpus.

28. The method as claimed in claim 27, wherein said step (g) further comprises the steps of:

(g2) determining n-gram statistics for each of said clustered audio segments in said audio segment corpus, wherein said n-gram statistics are based on a number of instances that a particular one of said clustered audio segments is contained in said audio segment corpus and are based on a number of instances that said particular one of said clustered audio segments is associated with at least another clustered audio segment in said audio segment corpus; and

(g3) calculating n-gram probability distributions of said n-gram statistics.

29. The method as claimed in claim 28, wherein said step (h) comprises the steps of:

(h1) determining n-gram segment statistic frequencies for said clustered audio segments based on said n-gram statistics for the clustered audio segments; and

(h2) determining said audio segment frequencies for said clustered audio segments based on said audio segment probability distributions.

30. The method as claimed in claim 29, wherein said step (i) comprises the steps of:

(i1) matching said audio segment frequencies to said known audio segment frequencies;

(i2) matching said n-gram segment statistic frequencies to known n-gram statistic frequencies for said at least one of letters, combination of letters, and words;

(i3) determining a best n-gram match of said n-gram segment statistic frequencies to said known n-gram statistic frequencies; and

(i4) determining a best segment match of said audio segment frequencies to said known audio segment frequencies based on said best n-gram match, wherein said best n-gram match and said best segment match at least partially constitute said frequency matches.

31. The method as claimed in claim 30, wherein said step (j) comprises the steps of:

(j1) comparing said audio segments corresponding to said audio segment frequencies of said best segment matcn with said audio similarity classes formed in said step (d);

(j2) determining which of said audio segments constitute matching audio segments that match said audio similarity classes; and

(j3) forming said textual corpus by using said matching audio segments, wherein said textual corpus maps certain audio similarity classes to certain audio segments.

32. An apparatus for training an automatic speech recognizer, comprising:

a receiver which inputs audio data;

a segmenting device which is coupled to said receiver and which segments said audio data to produce audio segments of said audio data;

an audio clustering device which is coupled to said segmenting device and which clusters said audio segments into groups of clustered audio segments, wherein said clustered audio segments in each of said groups have similar characteristics and wherein said groups respectively form audio similarity classes;

a probability calculating device which is coupled to said audio clustering device and which calculates audio segment probability distributions for said clustered audio segments in said audio similarity classes;

a frequency determining device which is coupled to said probability calculating device and which determines audio segment frequencies for said clustered audio segments in said audio similarity classes based on said audio segment probability distributions;

a frequency comparator which is coupled to said frequency determining device and which matches said audio segment frequencies to known audio segment frequencies for at least one of letters, combination of letters, and words to determine frequency matches;

a textual corpus generator which is coupled to said frequency comparator and which generates a textual corpus of words based on said frequency matches; and

an acoustic model trainer which is coupled to said textual corpus generator and which trains acoustic models of said automatic speech recognizer based on said textual corpus.

38. The apparatus as claimed in claim 37, wherein said probability calculating device further comprises:

an n-gram statistic determiner which determines n-gram statistics for each of said clustered audio segments in said audio segment corpus, wherein said n-gram statistics are based on a number of instances that a particular one of said clustered audio segments is contained in said audio segment corpus and are based on a number of instances that said particular one of said clustered audio segments is associated with at least another clustered audio segment in said audio segment corpus; and

an n-gram probability distribution calculator which is coupled to said n-gram

statistic determiner and which calculates n-gram probability distributions of said n-gram statistics.

39. The apparatus as claimed in claim 38, wherein said frequency determining device comprises:

an n-gram segment statistic frequency determiner which determines n-gram segment statistic frequencies for said clustered audio segments based on said n-gram statistics for the clustered audio segments; and

an audio segment frequency determiner which is coupled to said n-gram segment statistic frequency determiner and which determines said audio segment frequencies for said clustered audio segments based on said audio segment probability distributions.

40. The apparatus as claimed in claim 39, wherein said frequency comparator comprises:

an audio segment comparator which matches said audio segment frequencies to said known audio segment frequencies;

an n-gram segment comparator which matches said n-gram segment statistic frequencies to known n-gram statistic frequencies for said at least one of letters, combination of letters, and words;

a best n-gram match determiner which determines a best n-gram match of said n-gram segment statistic frequencies to said known n-gram statistic frequencies; and

a best segment match determiner which determines a best segment match of said audio segment frequencies to said known audio segment frequencies based on said best n-gram match, wherein said best n-gram match and said best segment match at least partially constitute said frequency matches.

41. The apparatus as claimed in claim 40, wherein said textual corpus generator comprises:

textual corpus comparator which compares said audio segments corresponding to said audio segment frequencies of said best segment match with said audio similarity classes formed by said audio clustering device;

a textual corpus determiner which determines which of said audio segments constitute matching audio segments that match said audio similarity classes; and

a textual corpus formation device which forms said textual corpus by using said matching audio segments, wherein said textual corpus maps certain audio similarity classes to certain audio segments.

Previous Doc Next Doc Go to Doc#

L6: Entry 2 of 3 File: USPT Jun 17, 2003

DOCUMENT-IDENTIFIER: US 6581057 B1
TITLE: Method and apparatus for rapidly producing document summaries and document browsing aids

Brief Summary Text (17):
In the specification and claims, the word "term" means single words, word n-grams, and/or phrases. An "n-gram" is a string of characters that may comprise all or part of a word.

Brief Summary Text (24):
The present invention also applies to document browsing aids, such as keyword gists, thumbnail images, clustering, and categories. A keyword gist is a shortened form of a document in which all but the keywords have been deleted. A thumbnail is a reduced image of the document (e.g., a photo reduction). Clustering involves grouping related documents together into a cluster. Categorizing is similar to clustering, but instead assigns a label to each document in which the label identifies which group that document belongs. By optimizing the search time generation of these aids through the precomputing and caching of information, the present invention makes these aids practical for real world applications, such as web catalogs and document indexes.

CLAIMS:

35. The computer-assisted method according to claim 32, wherein the query-biased document browsing aid is clustering.

L6: Entry 3 of 3 File: USPT Mar 8, 1994

DOCUMENT-IDENTIFIER: US 5293584 A
TITLE: Speech recognition system for natural language translation

Detailed Description Text (35):
The probability P(T) of occurrence of the target word sequence may be approximated by the product of n-gram probabilities for all n-grams in each string. That is, the probability of a sequence of words may be approximated by the product of the conditional probabilities of each word in the string, given the occurrence of the n-1 words (or absence of words) preceding each word. For example, if n=3, each trigram probability may represent the probability of occurrence of the third word in the trigram, given the occurrence of the first two words in the trigram.

Detailed Description Text (78):
The prototype vectors in prototype store 38 may be obtained, for example, by clustering feature vector signals from a training set into a plurality of clusters, and then calculating the mean and standard deviation for each cluster to form the parameter values of the prototype vector. When the training script comprises a series of word-segment models (forming a model of a series of words), and each word-segment model comprises a series of elementary models having specified locations in the word-segment models, the feature vector signals may be clustered by specifying that each cluster corresponds to a single elementary model in a single location in a single word-segment model. Such a method is described in more detail in U.S. patent application Ser. No. 730,714, filed on Jul. 16, 1991, entitled "Past Algorithm for Deriving Acoustic Prototypes for Automatic Speech Recognition."

Detailed Description Text (79):
Alternatively, all acoustic feature vectors generated by the utterance of a training text and which correspond to a given elementary model may be clustered by K-means Euclidean clustering or K-means Gaussian clustering, or both. Such a method is described, for example, in U.S. patent application Ser. No. 673,810, filed on Mar. 22, 1991 entitled "Speaker-Independent Label Coding Apparatus".

L3: Entry 2 of 4 File: USPT Jun 10, 2003

DOCUMENT-IDENTIFIER: US 6578032 B1
**** See image for Certificate of Correction ****
TITLE: Method and system for performing phrase/word clustering and cluster merging

Abstract Text (1):
Text classification has become an important aspect of information technology. Present text classification techniques range from simple text matching to more complex clustering methods. Clustering describes a process of discovering structure in a collection of characters. The invention automatically analyzes a text string and either updates an existing cluster or creates a new cluster. To that end, the invention may use a character n-gram matching process in addition to other heuristic-based clustering techniques. In the character n-gram matching process, each text string is first normalized using several heuristics. It is then divided into a set of overlapping character n-grams, where n is the number of adjacent characters. If the commonality between the text string and the existing cluster members satisfies a pre-defined threshold, the text string is added to the cluster. If, on the other hand, the commonality does not satisfy the pre-defined threshold, a new cluster may be created. Each cluster may have a selected topic name. The topic name allows whole clusters to be compared in a similar way to the individual clusters, and merged when a predetermined level of commonality exists between the subject clusters. The topic name also may be used as a suggested alternative to the text string. In this instance, the topic name of the cluster to which the text string was added may be outputted as an alternative to the text string.

Brief Summary Text (9):
Text classification has become an important aspect of information technology. Present text classification techniques range from simple text matching to more complex clustering methods. Clustering describes a process of discovering structure in a collection of characters. The invention automatically analyzes a text string and either updates an existing cluster or creates a new cluster. To that end, the invention may use a character n-gram matching process in addition to other heuristic-based clustering techniques. In the character n-gram matching process, each text string is first normalized using several heuristics. It is then divided into a set of overlapping character n-grams, where n is the number of adjacent characters. If the commonality between the text string and the existing cluster members satisfies a pre-defined threshold, the text string is added to the cluster. If, on the other hand, the commonality does not satisfy the pre-defined threshold, a new cluster may be created. Each cluster may have a selected topic name. The topic name allows whole clusters to be compared in a similar way to the individual clusters or strings, and merged when a predetermined level of commonality exists between the subject clusters. The topic name also may be used as a suggested alternative to the text string. In this instance, the topic name of the cluster to which the text string was added may be outputted as an alternative to the text string.

Brief Summary Text (10):
More specifically, the invention provides a method, system and computer-readable medium having computer-executable instructions for clustering character strings. Each character string comprises a word or a phrase. The method comprises the steps of receiving at least one character string, and clustering a first character string

with another character string into one or more groups, when the first character string satisfies a predetermined degree of commonality with one or more character strings in each of these groups. When the first character string does not satisfy the predetermined level of commonality with another character string, another group is created. The method also selects at least one of the character strings in each of the groups to be the group's topic name. Selection of the topic may be based on a pre-designation or a frequency of the received character strings with the groups. The selected topic may then be outputted.

Detailed Description Text (22):
Each cluster 306-309 may also designate at least one of its members to be a topic name. A topic name is one or more words or phrases that describe all members! of the cluster. Selection of a particular topic may be based on any number of factors including, but not limited to, the frequency with which a particular member is entered as a query and a predetermined user designation. In the example shown in FIG. 3, "pokemon" 300 is the topic for cluster A 306 because it is the only member of cluster A 306. However, if another of cluster A's 306 members, for example "pokeman" 301, was queried by users more often than "pokemon" 300, "pokeman" 301 may become the topic for cluster A 306. Alternatively, a database manager may predetermine that "pokemon" 300 will remain the topic for cluster A 306, regardless of the frequency of other queries. Selection of the topic will be discussed further with reference to FIG. 10.

Detailed Description Text (24):
Notably, the bigram character sets include spaces (i.e., "_") at the beginning and end of each word. This bigram segmenting is accomplished for received queries 301-304, as well as members of clusters 306-309. Although FIGS. 4-7 illustrate the comparison of received queries 301-304 with the members of clusters 306-309 using bigram matching, it should be appreciated that any n-gram matching may be conducted, for example, trigram or quadgram. It should also be appreciated that the invention may conduct the comparison of received queries 301-304 with the members of clusters 306-309 using other matching techniques.

Detailed Description Text (42):
In step 1004, QCluster Program 305 may calculate the frequency of the occurrence of the individual words and whole query. In step 1005, the highest frequency words and queries are determined, based on step 1004. The precise number of selected highest frequency "items" (i.e., words and/or queries) may vary, depending on the relative scores. For example, the two highest frequency items may be selected when their frequency scores are relatively close. On the other hand, only one highest frequency item may be selected, where the subject item has a frequency score that is significantly higher than the second highest frequency item. If two or more highest frequency items are selected, it is determined whether the items have the same frequency score, in step 1006. If the scores are not the same, the highest frequency item may be selected as the topic. Alternatively, a predetermined number of highest frequency items may be selected to be the topics. If the highest frequency items have the same frequency score, a predetermined criterion may be used to break the tie, in step 1008. For example, it may be that the longest item (i.e., the item with the most characters) is selected as the topic. Notably, if none of the items satisfy a predetermined minimum threshold to become a topic, it may be that the longest item is selected to be the topic of the cluster.

CLAIMS:

2. The method of claim 1, wherein said selection of said first and said another topic name further comprise determining a frequency of said words or phrases in said clusters, wherein said first and said another topic names satisfy a predetermined level of frequency in said clusters.

5. A method for classifying information, comprising: receiving at least one

character string, wherein each character string comprises a word or a phrase; segmenting a first character string into a first plurality of character sets and a another character string into another plurality of character sets, wherein each of said character sets comprise more than one adjacent characters of said character string; comparing said first plurality of character sets with said another plurality of character sets; clustering said first character string with said another character string into a group, when said first character set satisfies a predetermined degree of commonality with said another character set; creating another group when said first character set does not satisfy said predetermined degree of commonality with said another character set; selecting at least one of said character strings in each of said groups to be a topic, based on a frequency of said character strings with said groups; and outputting said topic.

Previous Doc Next Doc Go to Doc#